**校园社区平台**

作者姓名：赵安玉

专业名称：软件工程

指导教师：张玺 助教

**成都理工大学工程技术学院**

**学位论文诚信承诺书**

**本人慎重承诺和声明：**

1.本人已认真学习《学位论文作假行为处理办法》（中华人民共和国教育部第34号令）、《成都理工大学工程技术学院学位论文作假行为处理实施细则（试行）》（成理工教发〔2013〕30号）文件并已知晓教育部、学院对论文作假行为处理的有关规定，知晓论文作假可能导致作假者被取消学位申请资格、注销学位证书、开除学籍甚至被追究法律责任等后果。

2.本人已认真学习《成都理工大学工程技术学院毕业设计指导手册》，已知晓学院对论文撰写的内容和格式要求。

3.本人所提交的学位论文（题目：网上拍卖系统），是在指导教师指导下独立完成，本人对该论文的真实性、原创性负责。若论文按有关程序调查后被认定存在作假行为，本人自行承担相应的后果。

承诺人（学生签名）：

20 年 月 日

注：学位论文指向我校申请学士学位所提交的本科学生毕业

**摘要**

**关键词**：校园社区平台 SpringBoot GraphQL JavaWeb Vue

**Abstract**

目录

[摘要 I](#_Toc635723058)

[Abstract 2](#_Toc1369133069)

[前言 - 1 -](#_Toc1125898167)

[1 绪论 - 2 -](#_Toc1059961393)

[2 校园社区平台技术概述 - 2 -](#_Toc2089018456)

[2.1 概念 - 2 -](#_Toc628175011)

[2.1.1 基本概念 - 2 -](#_Toc1656478042)

[3 平台架构设计 - 3 -](#_Toc1131176229)

[3.1 开发技术及版本依赖 - 3 -](#_Toc1653377373)

[3.1.1 开发环境 - 3 -](#_Toc859484421)

[3.1.2 开发技术 - 3 -](#_Toc1914544919)

[3.2 平台架构 - 3 -](#_Toc608413784)

[3.2.1 平台后端模块概述 - 3 -](#_Toc756898537)

[3.2.2 平台前端概述 - 4 -](#_Toc1734575198)

[3.2.3 平台前后端通信技术 - 4 -](#_Toc1973594324)

[4 平台详细设计及实现 - 5 -](#_Toc149798315)

[4.1 平台数据存储设计 - 5 -](#_Toc2038664370)

[4.1.1 MySQL数据库表设计 - 5 -](#_Toc1129566413)

[4.2 平台后端设计与实现 - 8 -](#_Toc184803526)

[4.2.1 功能模块设计及概要 - 8 -](#_Toc412776091)

[4.2.2 部分功能设计与实现 - 18 -](#_Toc1424268980)

[4.3 平台前端设计与实现 - 18 -](#_Toc1911759956)

[4.3.1 前端技术选取分析 - 18 -](#_Toc749241873)

[4.3.2 页面设计与布局 - 18 -](#_Toc137806862)

[4.4 平台前后数据交互策略及实现 - 18 -](#_Toc42999170)

[4.4.1 交互技术选取技术分析 - 18 -](#_Toc982906996)

[4.4.2 数据通信详细设计 - 18 -](#_Toc135497281)

[4.5.3 数据交互详细实现 - 18 -](#_Toc511702305)

[5 测试与分析 - 19 -](#_Toc2084420925)

[5.1 测工具介绍 - 19 -](#_Toc1937477084)

[5.2 测试技术与方法概述 - 19 -](#_Toc1827336327)

[5.2 测试用例 - 19 -](#_Toc572660336)

[总结 - 20 -](#_Toc1159126505)

[参考文献 - 21 -](#_Toc805750846)

**前言**

**1 绪论**

# 2 校园社区平台技术概述

**2.1 概念**

### 2.1.1 基本概念

1. J2EE的概念

J2EE安全技术是一种通过使用了Java的平台和数据库来有效简化与企业和公司系统解决方案的设计和开发,增长和有效管理与业务有关的复杂安全问题的系统结构。

1. 模块化设计的概念和思想

模块化[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1/223952" \t "/home/anyu/Documents\\x/_blank)是指在进行[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1" \t "/home/anyu/Documents\x/_blank)时将一个大程序按照功能划分为若干小程序模块，每个小程序模块完成一个确定的功能，并在这些模块之间建立必要的联系，通过模块的互相协作完成整个功能的程序设计方法。在设计较复杂的程序时，一般采用自顶向下的方法，将问题划分为几个部分，各个部分再进行细化，直到分解为较好解决问题为止。模块化的目的是为了降低程序复杂度，使程序设计、调试和维护等操作简单化。

1. 前后端分离架构技术概念
2. MVC 设计模式概念
3. Spring 框架

Spring框架是[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0/3793459" \t "/home/anyu/Documents\\x/_blank)上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为[EJB](https://baike.baidu.com/item/EJB/144195" \t "/home/anyu/Documents\\x/_blank)（EnterpriseJavaBeans）模型的补充，甚至是替补。它简化了 Java 企业级应用开发。

1. SpingBoot 概念
2. MybatisPlus 框架概念
3. GraphQL API 查询语言的概念
4. JWT (Java Web Token)的概念

# 3 平台架构设计

**3.1 开发技术及版本依赖**

本节主主要介绍校园社区平台所用到的技术和开发环境，以及所依赖其他资源。

### 3.1.1 开发环境

操作系统：Linux 510

开发工具：IDEA 2020.3 Navicat

数据库：MySQL 8.0.20 Redis

容器技术：Docker

测试工具：Postman

SpringBoot 2.3.4.RELEASE

MybatisPlus 3.4.0

Java 11

Gradle 6.7

Vue 2

### 3.1.2 开发技术

**3.2 平台架构**

平台是一个前后端分离的web应用。后端使用了Gradle作为版本控制工具和模块管理工具；基于SpringBoot 的框架做快速开发; MybatisPlus做持久层框架,进行数据库操作。前端使用Vue框架搭建前端应用，利用Element 进行快速页面搭建。使用GraphQL作为前后端通信方式。

### 3.2.1 平台后端模块概述

各个模块遵循软件开发模块化设计原则，每个模块都只关注自己的功能或业务。其中ancf-msg、ancf-user、ancf-post业务模块按MVC模式实现，它们是和前端进行交互的模块；理论上 ancf-auth、ancf-oss、ancf-cache对外部模块只暴露业务接口，它们只负责对内部模块提供服务。

1、 ancf-webapp 模块，它是程序的主入口模块，最顶层模块；直接依赖于ancf-service-msg、ancf-service-user、ancf-service-post业务模块。程序的配置文件存放该模块中。

2、ancf-service-msg 模块，它是校园社区平台用户间消息发送、平台消息推送的业务实现。

3、 ancf-service-user 模块，它是校园社区平台与用户相关业务的实现，包括用户注册、用户登录、头像修改等等。

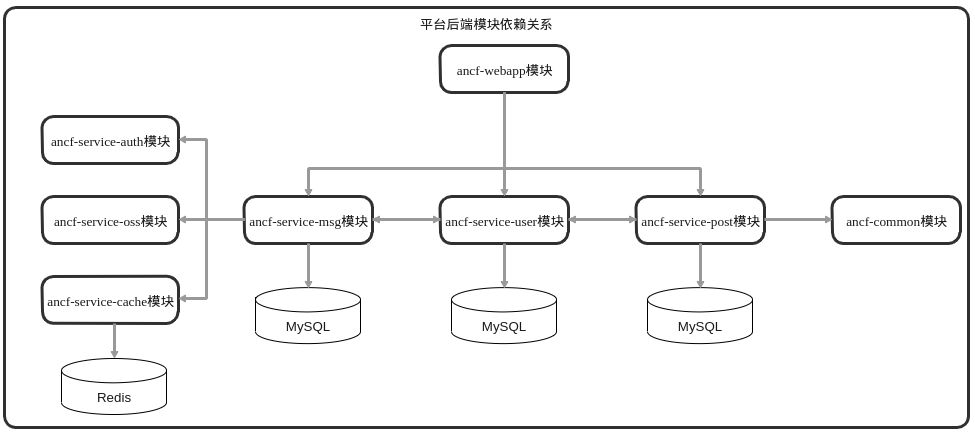
4、 ancf-service-post 模块，它是校园社区论坛与帖子相关业务的实现，包括发帖，评论、点赞等等。

5、ancf-common 模块，它是公共模块，最底层模块，基本上所有的模块都依赖于它。

6、ancf-service-auth 模块，它是平台权限认证、安全管理的实现。

7、 ancf-service-oss模块，它是平台文件上传云端服务器的实现，借助于阿里云OSS服务。

8、 ancf-sesrvice-cache 模块，它是系统缓存的业务实现。



### 3.2.2 平台前端概述

### 3.2.3 平台前后端通信技术

# 4 平台详细设计及实现

**4.1 平台数据存储设计**

关系型数据库使用MySQL，它作为平台主要的数据存储数据库，小部分实时热点数据或者有效期的数据使用Redis作为缓存数据库。

### 4.1.1 MySQL数据库表设计

对于实体表，提供了通用的字段表示必要信息。例如create\_time 创建时间、modified\_time修改时间、status状态信息。

1. 用户表 user，用户表包含平台用户的个人信息和认证信息以及一些账号状态信息。账号、id和邮箱都是唯一的，可用作登录。头像存储的是阿里云对象存储服务的图片地址。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | ID |
| account | char | 16 | NOT NULL | 账号 |
| gender | tinyint | 1 | NOT NULL | 性别 |
| age | tinyint | 3 | NOT NULL | 年龄 |
| birthday | datetime | - | NOT NULL | 出生日期 |
| nickname | char | 9 | NOT NULL | 昵称 |
| real\_name | char | 6 | NULL | 真实姓名 |
| avatar | char | 100 | NOT NULL | 头像存储地址 |
| email | char | 25 | NULL | 邮箱地址 |
| mobile | char | 11 | NULL | 手机号 |
| salt | char | 5 | NOT NULL | 密码加盐 |
| password | char | 50 | NOT NULL | 密码 |
| activation | tinyint | 1 | NOT NULL | 账号是否激活 |
| status | tinyint | 1 | NOT NULL | 用户状态 |
| create\_time | datetime | - | NOT NULL | 用户创建时间 |
| modified\_time | datetime | - | NOT NULL | 用户修改时间 |

1. 评论表 comment

评论表用于帖子的评论、评论的评论、或者评论的回复的表示，帖子的评论的字段entity\_type 必须为 1,target\_id 为null;评论的评论的字段entity\_type 必须为2，target\_id为null，评论的回复entity\_type 为3，target\_id 为null。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 评论id |
| content | varchar | 255 | NOT NULL | 内容 |
| user\_id | int | 10 | NOT NULL | 评论者id |
| entity\_id | int | 10 | NOT NULL | 评论的实体id |
| target\_id | int | 10 | NULL | 被评论者的id |
| entity\_type | tinyint | 1 | NOT NULL | 评论实体类型 |
| status | tinyint | 1 | NOT NULL | 评论状态 |
| create\_time | datetime | - | NOT NULL | 评论创建时间 |
| modifid\_time | datetime | - | NOT NULL | 评论修改时间 |

1. 帖子表 post

帖子的相关信息，

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 帖子id |
| type | tinyint | 1 | NOT NULL | 帖子类型 |
| user\_id | int | 10 | NOT NULL | 发帖者id |
| title | varchar | 20 | NOT NULL | 帖子标题 |
| content | text | - | NOT NULL | 内容 |
| cmt\_num | int | 10 | NOT NULL | 评论数 |
| score | float | - | NOT NULL | 得分 |
| status | tinyint | 1 | NOT NULL | 帖子状态 |
| create\_time | datetime | - | NOT NULL | 创建时间 |
| modified\_time | datetime | - | NOT NULL | 修改时间 |

1. 消息表 message

私信或者平台内部消息的实体、

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 消息id |
| from\_id | int | 10 | NOT NULL | 发送者id |
| to\_id | int | 10 | NOT NULL | 接受者id |
| chart\_id | varchar | 20 | NOT NULL | 聊天id |
| content | varchar | 255 | NOT NULL | 内容 |
| status | tinyint | 1 | NOT NULL | 状态 |
| create\_time | datetime | - | NOT NULL | 创建时间 |
| modified\_time | datetime | - | NOT NULL | 修改时间 |

**4.1.2 Redis缓存设计**

1. 设计缘由以及问题分析

考虑到部分数据修改和查询频繁，MySQL数据库压力过大，和程序效率问题

一部分数据具有时效性，过期的数据没有任何可利用价值。

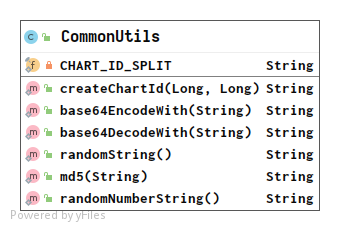
**4.2 平台后端设计与实现**

### 4.2.1 功能模块设计及概要

**1、ancf-common**

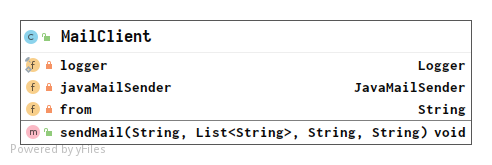
1. CommonUtils 平台公共工具类

提供给其他模块的公用工具类，里面包括了base64加密解密、生成随机字符串、生成随机数字、md5加密。

****

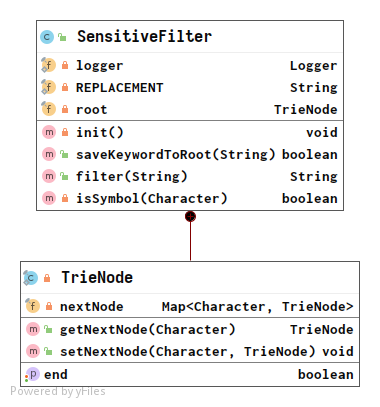
1. MailClient 系统邮件发送组件类

邮件客户端使用了springboot自带的邮件依赖，对其进行了自定义封装，通过调用sendMail方法发送邮件，可以一次分发多条邮件信息。

****

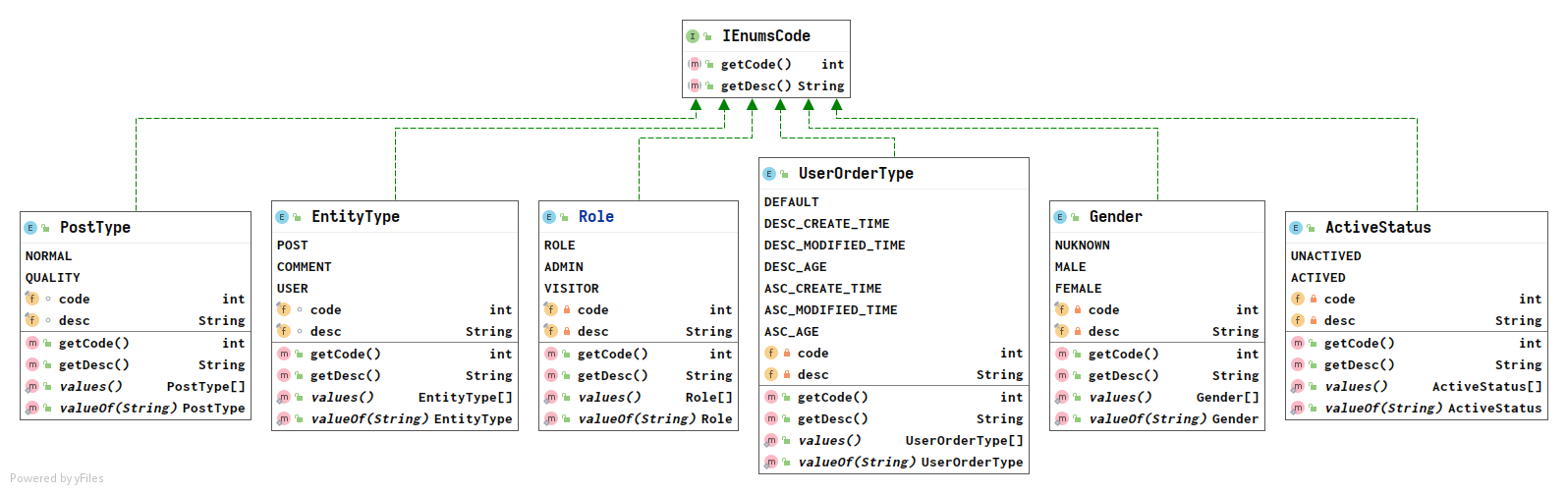
1. 平台敏感字文本文件过滤类

对于一个一个公共的平台，需要对敏感信息进行过滤处理，SensitiveFilter 正是平台的敏感词过滤处理工具类。其主要实现借助于字典树这一数据结构。init方法在初始化时会读取平台的敏感字文件，将每一个单词通过saveKeywordToRoot 方法加入到敏感字字典树中。最后生成一个有所有敏感字信息的字典树。filter方法对传入的文本文件结合已生成的敏感字字典树进行算法处理，最后返回过滤的文本。

****

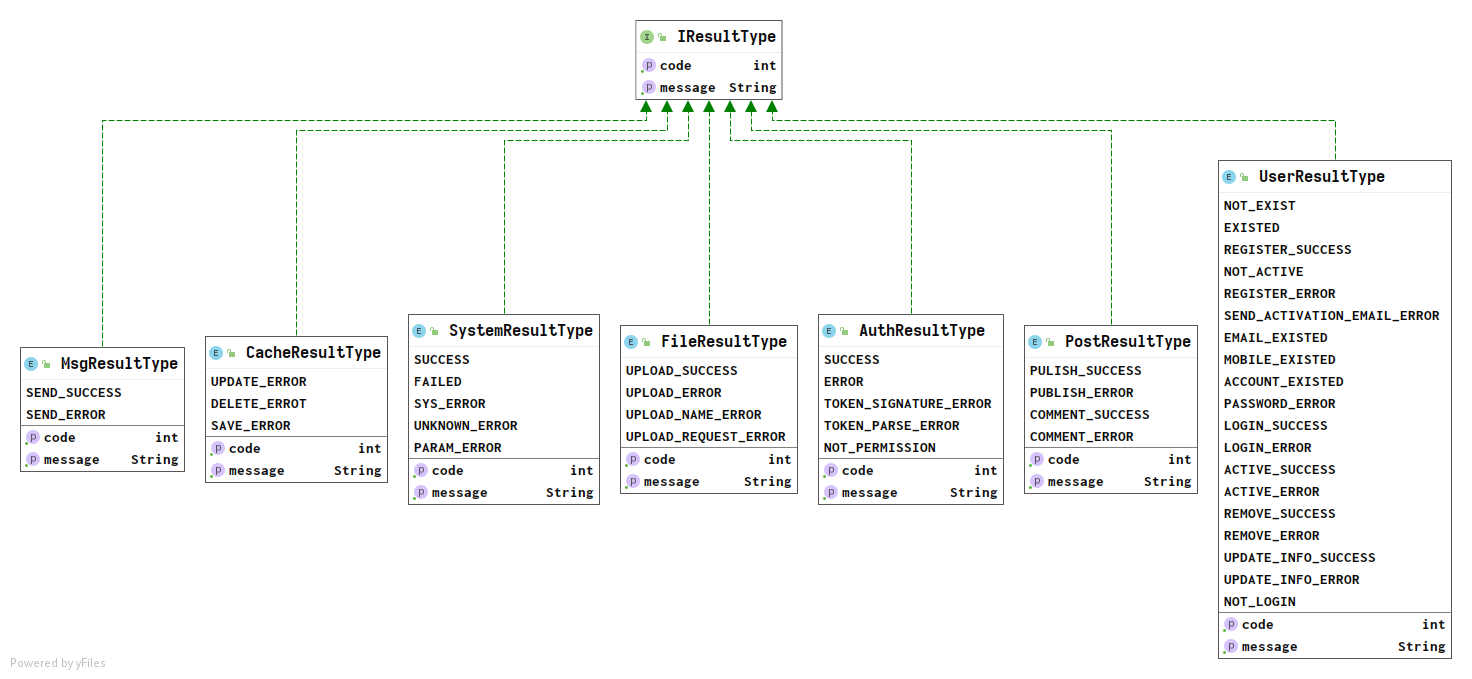
1. 数据库字典枚举类实现

数据库的某一些字段表示的值或者内容范围较小且固定，在设计数据库这些字段时，考虑到性能与合理性，通常字段类型使用tinyint类型；与Java中的Integer相互对应，考虑到数字所表示的信息在实际代码开发中对开发人员不够清晰，因此使用枚举俩定义这些字段；本平台的数据库字典枚举都继承IEmnuCode接口，通过实现该接口用来表示这是数据库字典，getCode 方法方法获取该枚举所存入数据库的值，getDesc 方法获取该枚举的描述信息。

****

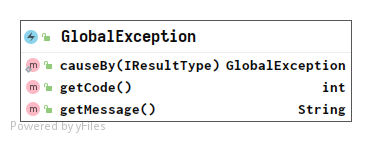
1. 业务处理结果状态枚举类

考虑到任何有意义的请求，服务器端都会进行业务操作，最终得到有且只有一个结果状态，返回给客户端。因此平台设计对此类结果状态进行统一化，规范化。所有业务处理结果的状态的枚举都实现IResultType接口，该接口有且只有三个方法，分别是isSuccess方法、 getCode方法 和getMessage方法；isSuccess方法表示业务处理结果是否成功，getCode方法返回业务处理结果的业务代码，getMessage方法返回业务处理结果的业务结果信息。

****

1. 自定义全局异常类

这是本平台所自定义的异常类继承于RunTimeException异常类，添加了code属性，重写了getMessage方法，返回业务结果状态信息。提供了一个静态 causeBy方法，用于通过业务处理结果状态，生成异常信息。

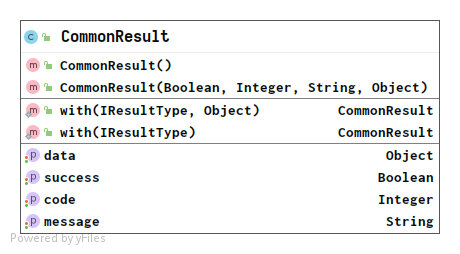


1. 全局异常处理，实现GraphQLErrorHandler接口，重写了processErrors方法。只返回给浏览器业务处理错误信息和业务代码。



1. 统一结果返回

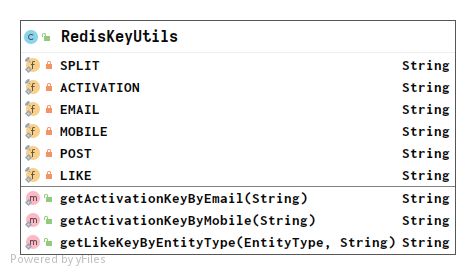
用于mutation类型的数据结果返回给客户端。提供了静态的with方法，可通过传入业务处理结果状态枚举类，来生成统一结果。



2、ancf-service-cache模块

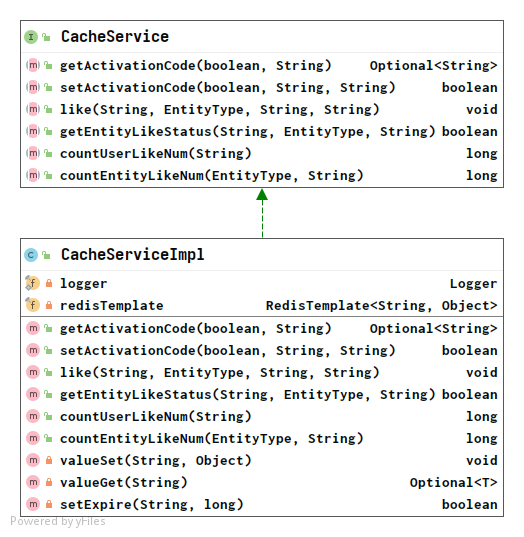
1. Redis 键生成工具类

缓存信息时，需要根据某些信息生成特定的键，方便再次从缓存中获取缓存信息。



1. 缓存服务接口及实现

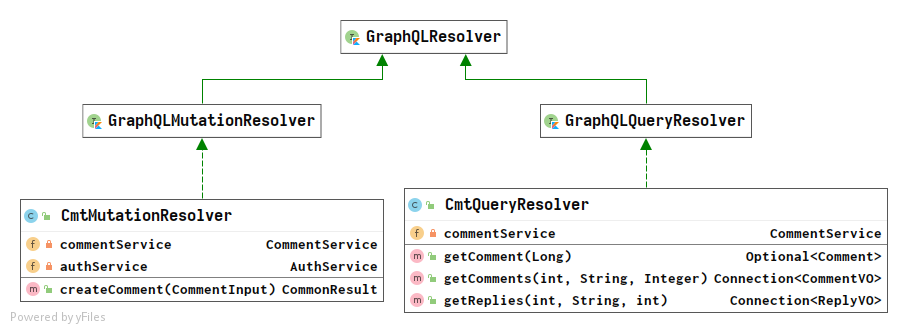
其他模块使用缓存是只能通过调用缓存接口，进行缓存功能的实现。目前系统对账号激活的激活码、帖子的点赞数和点赞状态进行缓存。

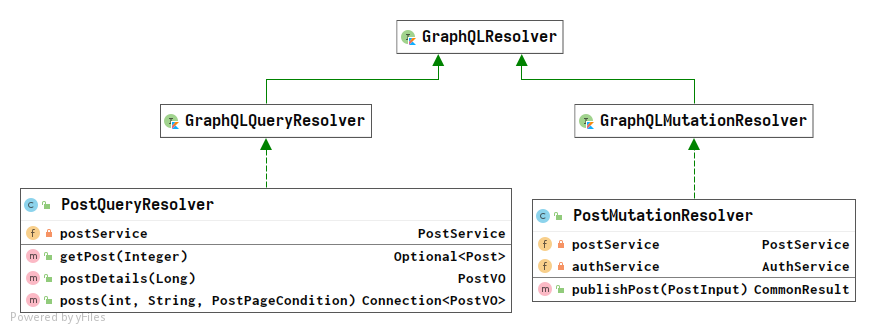
****

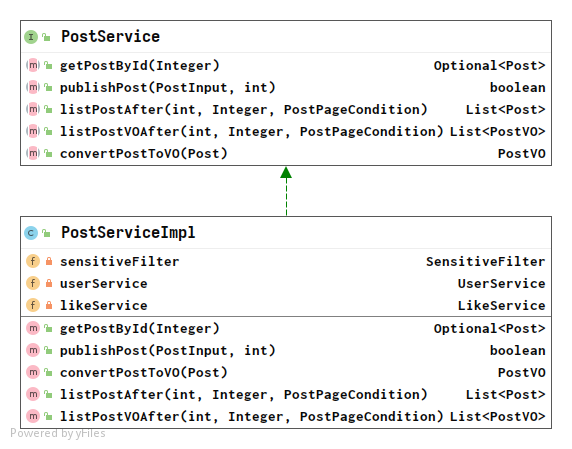
3、ancf-service-msg 模块

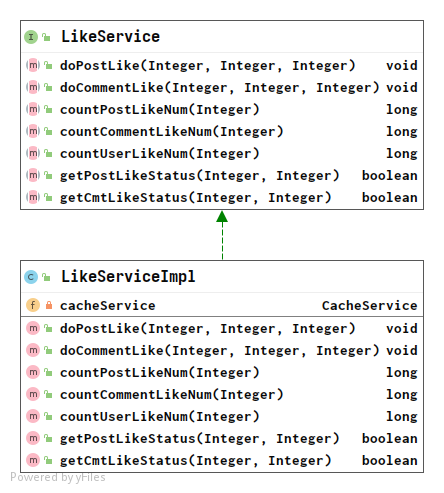
4、ancf-service-post 模块

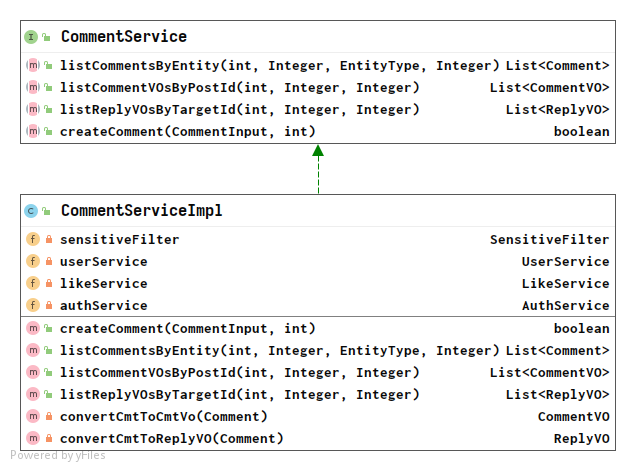
1. 帖子API数据交互层,实现GraphQLMutationResolver接口。为数据变更操作，

****

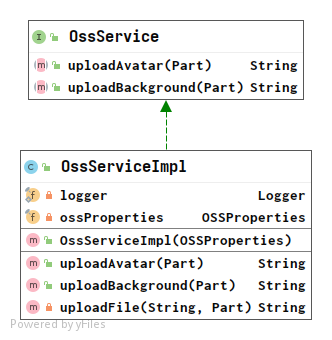
****

****

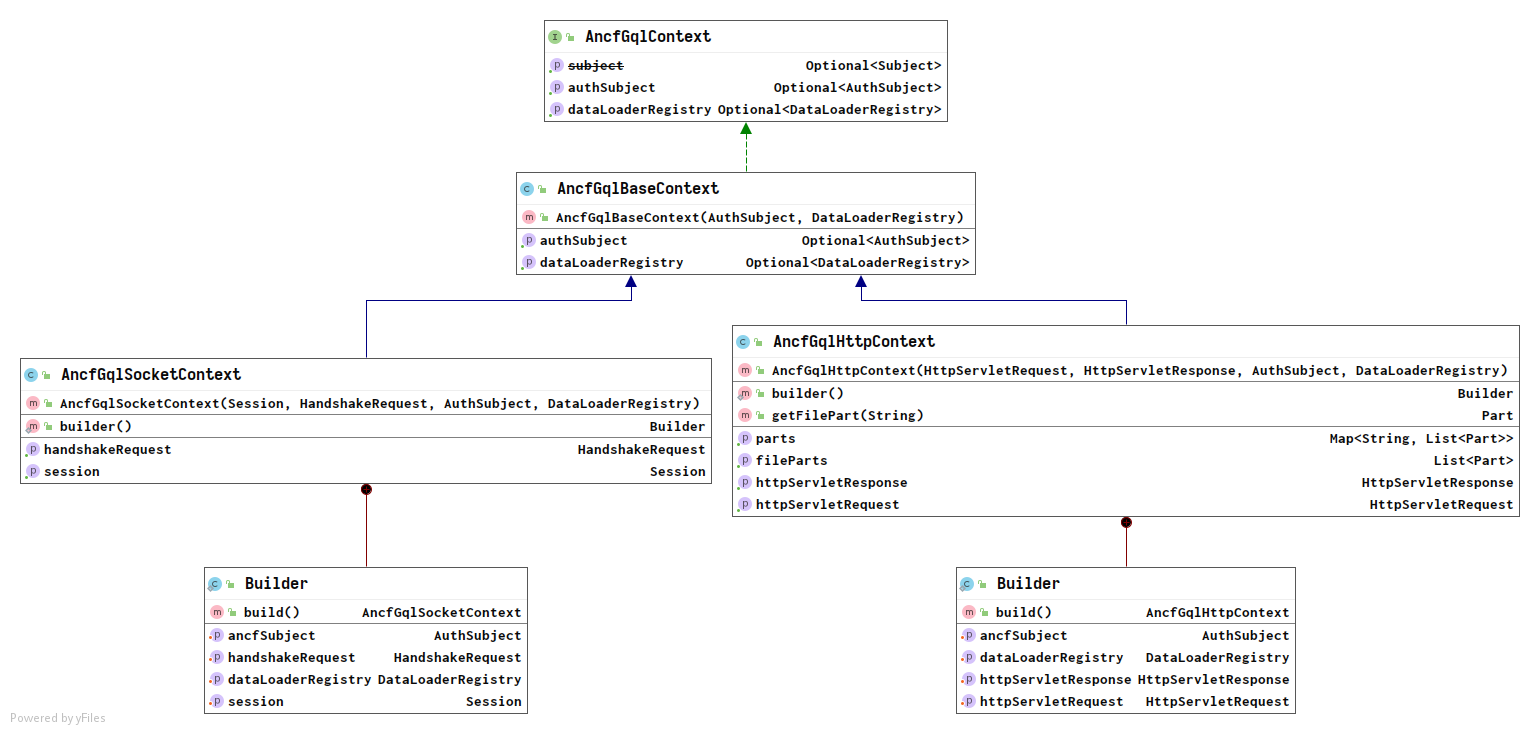
****

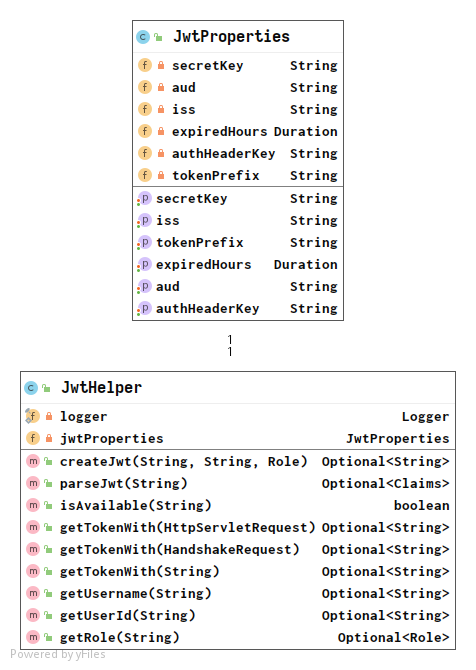
****

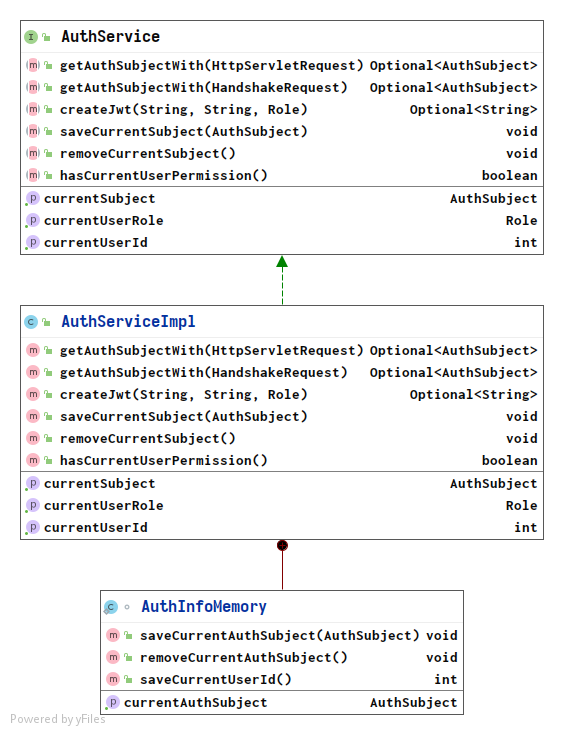
5、ancf-service-oss模块



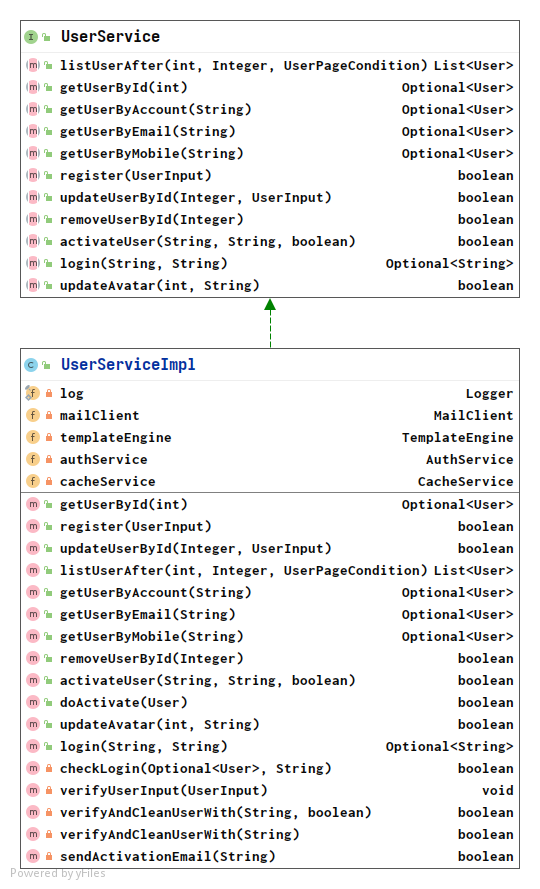
6、ancf-service-auth模块

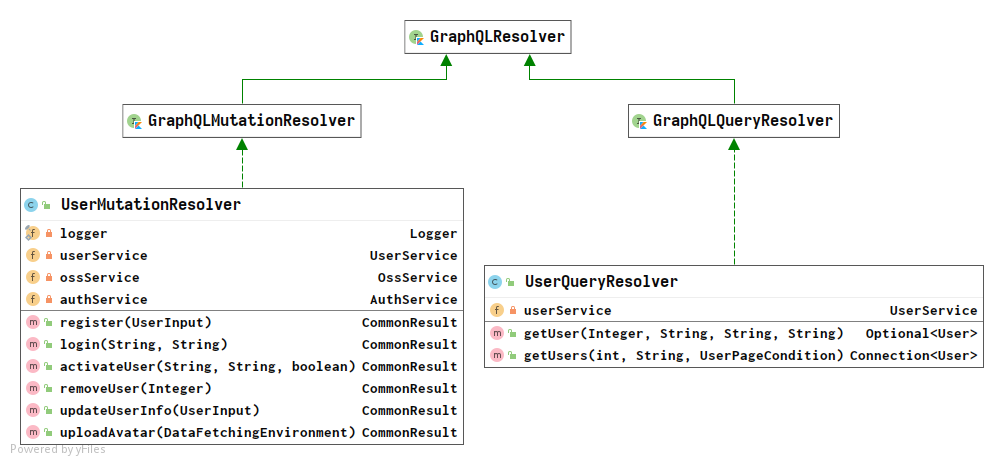
****

****

****

**7、**ancf-service-user模块

****

****

**8、**ancf-webapp模块

### 4.2.2 部分功能设计与实现

**4.3 平台前端设计与实现**

### 4.3.1 前端技术选取分析

### 4.3.2 页面设计与布局

**4.4 平台前后数据交互策略及实现**

### 4.4.1 交互技术选取技术分析

### 4.4.2 数据通信详细设计

### 4.5.3 数据交互详细实现

# 5 测试与分析

**5.1 测工具介绍**

**5.2 测试技术与方法概述**

**5.2 测试用例**

**总结**

**参考文献**